

IN THE CLAIMS:

1. (Currently amended) A modular assembly of software configured to operate hardware ~~perform a function~~, said modular assembly comprising:

a plurality of atoms, each of said plurality of atoms designed to execute a defined task;

a plurality of maps invoking a portion of said plurality of atoms for executing events that include a portion of said defined ~~tasks~~ task; and

a map engine in communication with each of said plurality of maps, said map engine coordinating an order and a timing for starting of each of said plurality of maps, wherein said map engine modifies said order and said timing based on inputs and variables received thereby before and during operation of said plurality of maps to operate or modify the operation of the hardware.

2. (Original) A modular assembly as set forth in claim 1 wherein said map engine includes a prioritizer to identify said order and said timing of execution of each of said plurality of maps and each of said plurality of atoms.

3. (Original) A modular assembly as set forth in claim 2 wherein said prioritizer includes input lines which receive inputs from clients that may change said order and said timing of execution of each of said plurality of maps.

4. (Original) A modular assembly as set forth in claim 3 wherein each of said plurality of atoms includes a design element classifying a type of executable, identifying inputs required to operate the executable and identifying a purpose therefor.

5. (Original) A modular assembly as set forth in claim 4 wherein each of said plurality of atoms includes an execution element that executes said defined task.

6. (Currently Amended) A modular assembly of software configured to operate hardware ~~perform a function~~, said modular assembly comprising:

a plurality of atoms designed to execute a plurality of tasks, each of said plurality of atoms including a design element and an execution element such that each of said

design elements identifies a type of executable, inputs required by each of said plurality of atoms and purpose therefor, and each of said execution elements execute said defined tasks task;

a plurality of maps invoking a portion of said plurality of atoms for executing events that include a portion of said defined ~~tasks~~ task; and

a map engine in communication with each of said plurality of maps, said map engine coordinating an order and a timing for starting of each of said plurality of maps, wherein said map engine modifies said order and said timing based on inputs and variables received thereby before and during operation of said plurality of maps to operate or modify the operation of the hardware.

7. (Original) A modular assembly as set forth in claim 6 wherein said map engine includes a prioritizer to identify said order and said timing of execution of each of said plurality of maps and each of said plurality of atoms.

8. (Currently Amended) A method of operating hardware, using performing a function, including a plurality of defined tasks, ~~using~~ a map engine, a plurality of maps, and a plurality of atoms, each having design and executable elements, the method comprising the steps of:

activating the map engine;
cataloging each of the plurality of atoms so that the map engine has an accurate inventory of the plurality of atoms available;
identifying an occurrence of an event;
associating the event with one of the plurality of maps;
loading the one of the plurality of maps associated with the event; and
executing each of the plurality of atoms identified with the one of the plurality of maps such that the operation of the hardware ~~function to be performed~~ is done in response to the occurrence of the event.

9. (Original) A method as set forth in claim 8 including the step of loading a plurality of maps, each being loaded in response to an identification of an event.

10. (Original) A method as set forth in claim 9 wherein the step of identifying an occurrence of an event includes the receipt of an input.

11. (Original) A method as set forth in claim 10 including the step of receiving the input into one of the plurality of atoms.

12. (Original) A method as set forth in claim 11 including the step of changing the plurality of atoms associated with the plurality of maps based on a change in the function to be performed.

13. (Original) A method as set forth in claim 12 including the step of changing the plurality of maps associated with function based on a change in the function to be performed.

14. (Original) A method as set forth in claim 13 wherein the step of changing the plurality of atoms includes changing the number of atoms being executed.

15. (Original) A method as set forth in claim 14 wherein the step of changing the plurality of maps includes changing the number of maps being invoked.

16. (Original) A method as set forth in claim 15 wherein the step of changing the plurality of atoms is performed independently of the step of changing the plurality of maps.

17. (Original) A method as set forth in claim 16 wherein the step of changing the plurality of maps is performed independently of the step of changing the plurality of atoms.

18. (Original) A method as set forth in claim 17 wherein the step of changing the plurality of atoms includes the step of modifying the number of the plurality of atoms being executed with one of the plurality of maps.

19. (Original) A method as set forth in claim 18 wherein the step of changing the plurality of atoms includes the step of modifying an order in which the plurality of atoms are executed within one of the plurality of maps.